Chapter Two: Contents

(Population Synthesizer – 31 August 2004 – LA-UR 00-1725 – TRANSIMS 3.1)

Volume Four: Figures

Volume Four: Tables

1. Introduction

1.1 Overview

The Urban Infrastructure Suite (UIS) methodology is based on the movement of individual travelers between activities at different locations; therefore, the Population module must create a synthetic population that represents every individual in the metropolitan region under study. Population demographics are crucial in creating reality-based simulations because such demographics determine the level of activity for each household.

Demographic examples include the individual's age, income, gender, and employment status. Such demographics determine how each individual travels across the transportation network. For example, a six-year-old girl will take the bus to school, whereas a 30-year-old executive will carpool to work.

The procedures outlined in this chapter generate a baseline year 2000 synthetic population and provide methods that can update this population to a future year.

1.2 Source Data

To create a virtual population, the Population Generator requires the following types of source data:

- U.S. Census Bureau Summary File 3 (SF 3) data[1],
- U.S. Census Bureau Public Use Microdata Sample (PUMS) [2],
- Master Area Block Level Equivalency/Geographic Correspondence Engine (MABLE/Geocorr2k), and
- Forecast marginal demographic data.

1.2.1 SF 3 Data

These files contain demographic summary tables from the 2000 Census for small geographic areas, census tracts, or census block groups. Mostly one-dimensional, these summary tables contain information such as the distribution of the age of the householder or the number of workers in the family. Table 1 and Table 2 show typical SF 3 data.

[1] U.S. Census Bureau, 2000 Census of Population and Housing, Summary File 3: Technical Documentation, 2002.
[2] U.S. Census Bureau, 2000 Census of Population and Housing, Public Use Microdata Sample, United States: Technical Documentation, 2003.

Table 1. Number of workers in family households for census tract 1, block group 2 of Los Alamos County, NM.

	Number of Households, *n*, with Number of Workers in Household			
Workers	0	1	2	>2
n	0	121	214	25

Table 2. Age distribution of householders for census tract 1, block group 2 of Los Alamos County, NM.

	Number of Households, *n*, with Householder Age in the Given Ranges						
Age	15-24	25-34	35-44	45-54	55-64	65-74	>74
n	4	134	94	46	46	36	0

1.2.2 PUMS

Census tracts or block groups that are combined into much larger geographic areas are known as Public Use Microdata Areas, or PUMA. Microdata in each PUMA, the PUMS, consist of a 5% sample of the complete census records for the PUMA. PUMS contain the complete structure of each household, including the number of people in a given household, the household income, number of workers, and number of vehicles owned. These files are edited to protect the confidentiality of all individuals, but they have the information necessary to conduct effective research and analysis.

1.2.3 MABLE/Geocorr2k

Our methodology requires that the census block information obtained from SF 3 be correlated with the PUMS. SF 3 contains fields for PUMA information but, unfortunately, these fields are blank in the data files.

The MABLE/Geocorr2k database search engine accesses data maintained by the University of Missouri's Office of Social and Economic Data Analysis in cooperation with the Missouri Census Data Center. The data and search engine are available at the following Internet site:

http://mcdc2.missouri.edu/websas/geocorr2k.html

This tool may be used to obtain a correspondence between PUMAs and census block groups.

1.2.4 Forecast Marginal File

When creating a forecast population, the Population Generator requires a "Forecast Marginal File" as input. This file contains the forecast marginal distributions, similar to those given above in Table 1 and Table 2, as a function of census tract and block group. Forecasts can typically be obtained by transportation planning agencies, and this information is then compiled into the format required by the Population Generator.

1.3 Using the Data

The Population Generator implements an algorithm developed by Beckman, Baggerly, and McKay[3] to generate synthetic populations from the four data sets described above. Land-use data are used to place individual households at activity locations along the transportation network. A complete description of the algorithm appears in Section 4 (*Algorithm*).

[3] Beckman, Richard J., Baggerly, Keith A., and McKay, Michael D. (1996), Creating Synthetic Baseline Populations, *Transportation Research* A, Vol 30, No. 6, pp 415-429.

2. MODULE DESCRIPTION

2.1 Overview

Fig. 1 shows the types of data the Population Generator uses to generate a synthetic population of households that contain individual demographics and household locations within the transportation network.

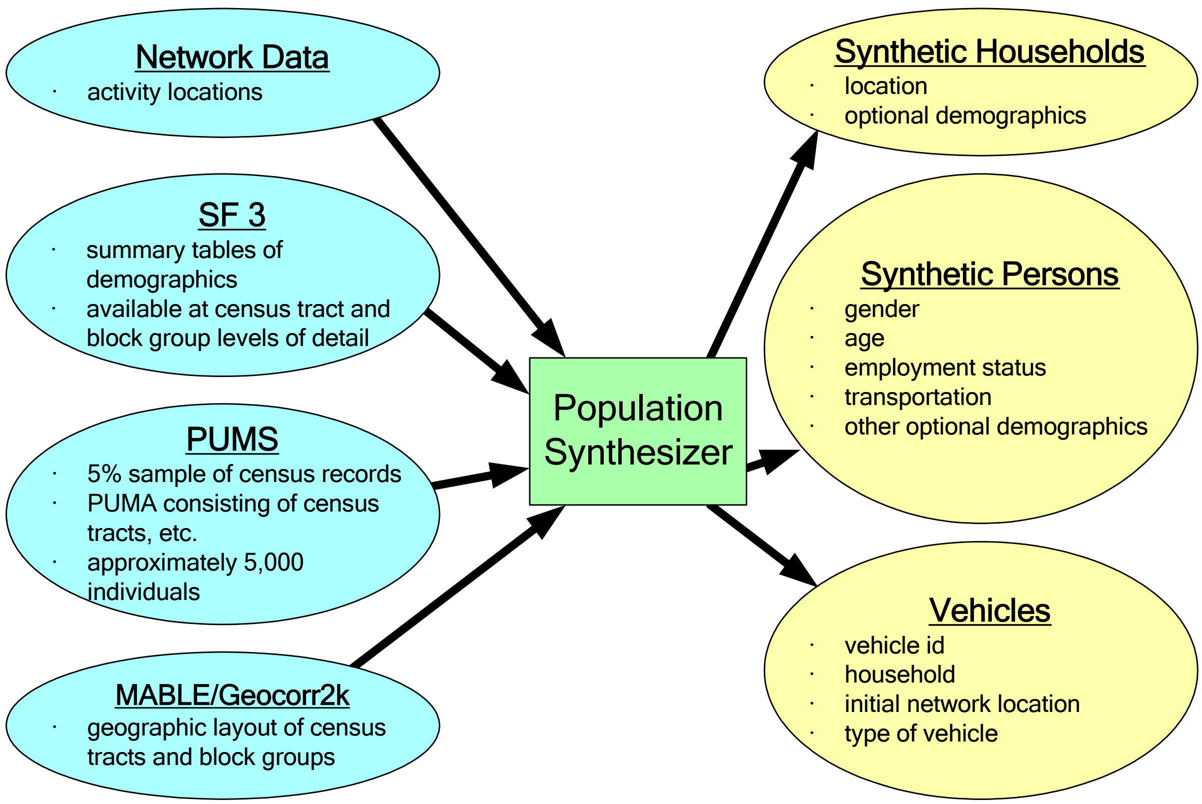


Fig. 1. The Population Generator takes in various types of census data to generate synthetic households, individuals, and vehicles.

2.2 Defining a Household

In the baseline methodology presented here, each household in a synthetic population is composed of all of the people who occupy a housing unit. This could be a single family, one person living alone, or a group of people living together. People living in group quarters are not considered at this time. Household demographics vary in accordance with source data and study needs.

2.2.1 Using Households

The Population module assigns households to activity locations on a link of the transportation network. Each activity location is assigned to a link on the network and is associated with the land-use characteristics that surround it. Multiple households can be assigned to one activity location.

Example**:**

> An activity location on a link can represent one side of a street, whereas a second could represent activities taking place on the other side. Moreover, activity locations could represent individual buildings on a street, or one activity location could represent all activities that take place on the street.

Synthetic populations in essence "drive" the Activity Generator, which uses the data to create individual travel activities, such as work, school, or shopping. Population data also can be used to categorize and filter population subsets used for various types of equity analyses.

2.3 Data Format

Any viable source of household and demographic data can be used to construct a synthetic population, provided that the output of the computation is formatted in accordance with the required data formats for synthetic populations.

3. Generating a Synthetic Population

Step One

- Identify an appropriate PUMA.
- Use MABLE/Geocorr2k to obtain a list of block groups within the PUMA.

Step Two

Obtain summary statistics from SF 3 for each of the block groups identified in the PUMA.

Example:

Summary data could include the householder's age, the household income, the household size, and the number of workers in the household.

Step Three

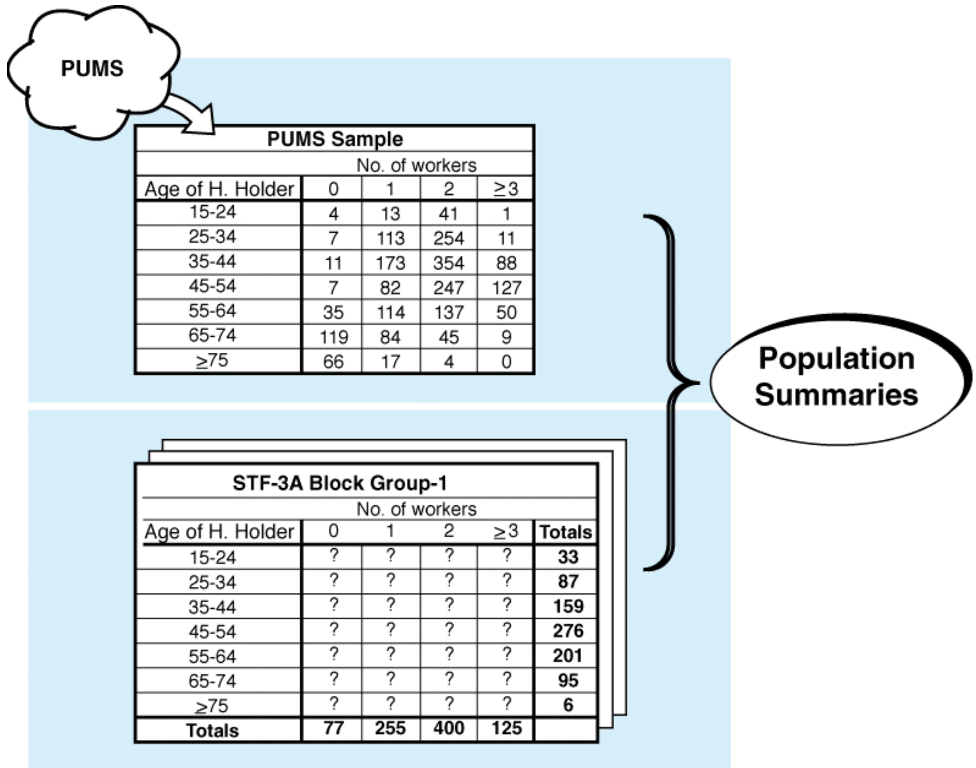
- Construct a multidimensional table from the PUMS data.
- Make sure that the dimensions correspond with SF 3 summary statistics.

Note: Fig. 2 represents these steps in graphic form.

Example:

In Fig. 2, the multidimensional table would have four dimensions that correspond to four classifications: householder's age, the household's income, household size, and number of workers in the household.

- Each household in the PUMS has a household weight. The sum of these weights for each of the households in the PUMS for each classification compose the multidimensional table.



PUMS Sample

Age of H. Holder	No. of workers 0	1	2	≥3
15-24	4	13	41	1
25-34	7	113	254	11
35-44	11	173	354	88
45-54	7	82	247	127
55-64	35	114	137	50
65-74	119	84	45	9
≥75	66	17	4	0

STF-3A Block Group-1

Age of H. Holder	No. of workers 0	1	2	≥3	Totals
15-24	?	?	?	?	**33**
25-34	?	?	?	?	**87**
35-44	?	?	?	?	**159**
45-54	?	?	?	?	**276**
55-64	?	?	?	?	**201**
65-74	?	?	?	?	**95**
≥75	?	?	?	?	**6**
Totals	**77**	**255**	**400**	**125**	

Fig. 2. Various data types are used to generate a synthetic population. Note that these data have been modified to ensure anonymity.

Step Four

- At this point, the proportion of households for each block group's classification is unknown. To determine this, we use a two-stage iterative proportional fitting procedure outlined by Beckman, Baggerly, and McKay[4].
- This procedure satisfies the distributions of the SF 3 data for each block group while maintaining the correlation structure of the table constructed from the PUMS.

Step Five

- The block groups are updated for a forecast.
- Iterative proportional fitting uses the correlation structure of the generated block group demographic tables and the SF 3 type forecast demographics for the update.

Step Six

- Select households from the PUMS to match the number of households in the Census over a given geographic area, such as a block group or a census tract.

[4] See Footnote 3.

- Use land-use information to place each household within a block group at an activity location.

  Note**:** A baseline population for the census year can be generated by skipping Step Five.

  Note**:** Land-use data are stored in the network activity location files. At a minimum, these files contain the identity of the activity locations, their locations, the corresponding block group and census tract, and some indication of the activities that may be performed at that location.

  Fig. 3 shows this step graphically.

- Identify the activity locations within a block group before placing households from the synthetic population at an activity location.

- Using land-use data, determine a weight for each activity location. These weights are proportional to the probability that a household will be placed at the activity location. The weights could be formed, for example, by adding the single family residential square footage to the multiple of the multi-family square footage for each activity location. In another approach, the number of households on a block could be determined from phone books and used as the weights.

- Divide each individual weight for an activity by the total weight of all the activity locations in the block group.

- The resultant ratios are used as the probability of a household being located on a link.

- For each synthetic household, a random activity location (based on the probabilities) is selected; the household is then placed at that activity location.

  Note**:** Households need not be placed at unique activity locations. Many households can share the same activity location.

- The household location algorithm can remain the same no matter which area is being studied. However, the weights given to the activity locations in a block group will depend on the quality and availability of land-use data.

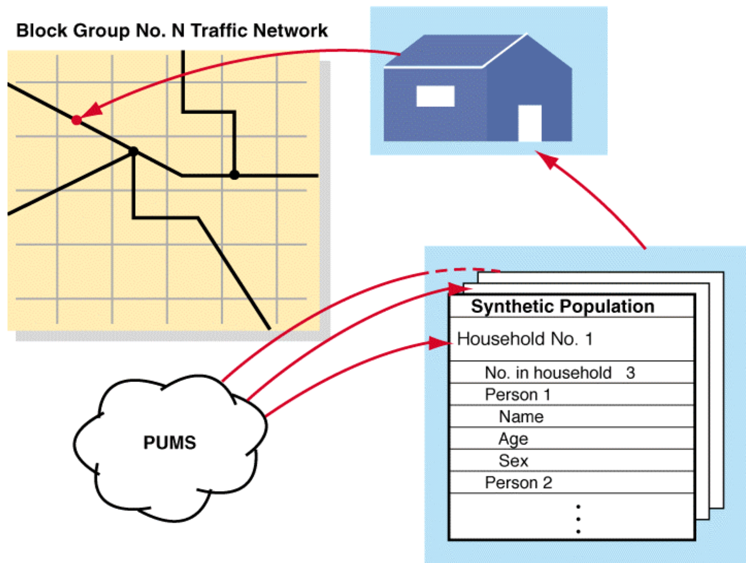


Fig. 3. Creating households and placing them on the network involves randomly selecting actual PUMS households in accordance with the proportions derived from the Iterated Proportional Fitting method.

3.1 Other Techniques

Households may be located on a link by other techniques.

Example 1:

A census block could be used to determine the number of households in a block. This number could then be associated with an activity location and used as the weight.

Example 2:

Electronic phone books or aerial photography could be used to determine the number of households in a block.

Note**:** To date, neither of these techniques has been used during a case study.

3.2 Vehicle Ownership

To assign vehicle ownership, we currently use either the number generated from the synthetic population procedure based on the PUMS or simply assign a vehicle to every possible driver. A more refined vehicle ownership model based on population demographics and network characteristics could be implemented and applied after the synthetic population has been generated.

In traditional methods, household vehicle ownership has been a factor in the choice of transportation modes. Given the iterative methods used in UIS to determine the traveler's mode of transportation using the iteration database and the Selector, a refined vehicle ownership model is not necessary.

In either case, each vehicle represents an entry in a vehicle file. This file contains a vehicle identification number, the household to which it is assigned, and the vehicle emissions type. The emissions type is used in the Emissions Estimator module to determine emissions. It reflects the operating condition of the vehicle, its type, and age.

Presently, these vehicle types are assigned at random according to a national or local distribution of 23 vehicle emission types described in Volume Three (*Modules*), Chapter Seven (*Emissions Estimator*). In the future, an analyst may wish to use a vehicle type model based on Department of Motor Vehicle statistics, inspection and maintenance records, and the synthetic population.

4. ALGORITHM

4.1 Overview

Fig. 4 shows the data flow used to create, locate, and assign vehicles to a complete synthetic population of households and individuals.

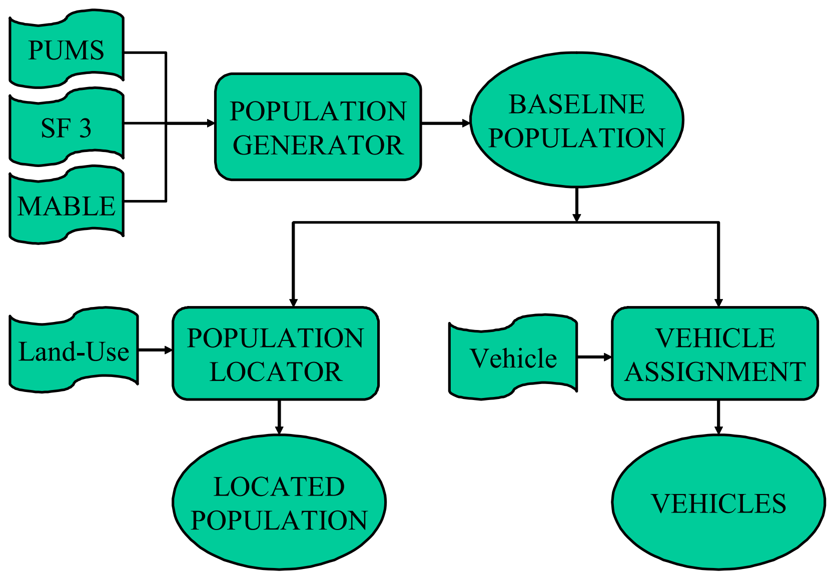


Fig. 4. This flowchart shows the process used to create, locate, and assign vehicles to a complete synthetic population and individuals.

4.2 Baseline Population Generator

Of the processes shown in Fig. 4, the baseline Population Generator is the most complex. Its objective is to create a population over small geographic areas of census groups that maintain the statistical characteristics of the census. However, as shown in Table 3, the summary data for block groups from SF 3 do not give the entries for any cross-classified demographics.

Table 3. The cross-classification of the number of workers and the age of the householder for census tract 1, block group 2 of Los Alamos County, NM, is unknown.

	Householder Age							
Workers	15-24	25-34	35-44	45-54	55-64	65-74	>74	Total
0	?	?	?	?	?	?	?	0
1	?	?	?	?	?	?	?	121
2	?	?	?	?	?	?	?	214
>2	?	?	?	?	?	?	?	25
Total	4	134	94	46	46	36	0	

A synthetic population is easily generated if cross-classified tables exist for small areas such as block groups. Because PUMS contains complete household records, these could be drawn at random, thus satisfying the cross-classified table for the block group. This is the general scheme for the algorithm presented in this document, except that the cross-classified table for the block groups is estimated. This estimation process satisfies the totals as given by SF 3.

The general methodology consists of three steps.

Step One Select a reasonable set of demographics from SF 3 that characterize the population.

Step Two For each block group, estimate the proportions in the cross-classified table made up of the demographics selected in Step One.

Step Three Draw households at random from the PUMS corresponding to the block group so that the estimated proportions in the cross-classified table are satisfied.

4.3 Multiway Summary Tables

Although cross-classified tables cannot be derived from SF 3 for small areas, multiway summary tables can be created for the entire PUMA area. For example, block group 2 of census tract 1 for Los Alamos County, NM, is contained in PUMA 00400. Table 4 provides the multiway table for this PUMA. It shows the number of workers in a family and the age of the householder.

Table 4. The cross-classification of the number of workers and the age of the householder for PUMA 00400, which contains census tract 1, block group 2 of Los Alamos County, NM.

Householder Age								
Workers	15-24	25-34	35-44	45-54	55-64	65-74	>74	Total
0	2	11	9	3	26	64	42	157
1	11	108	122	48	80	61	18	448
2	28	135	274	156	85	22	6	706
>2	0	3	65	76	40	10	3	197
Total	41	257	470	283	231	157	69	

To estimate the proportions in the cells of the multiway block group tables, we use iterative proportional fitting[5] (IPF) of the block group summaries to the cross-classified values in the PUMS. IPF ensures that the correlation structure of the demographics for every entity that contributes to the PUMA (e.g., block groups) is the same as the correlation structure in the multiway tables constructed from the PUMS.

IPF assumes that we have

1) a sample from a multiway classification of characteristics, and

2) the exact totals for the margins of the multiway table.

In this case, we could assume that the PUMS represents the sample and the SF 3 data give the marginal totals. We show later that this is an oversimplified view of these data, but we continue with this to better explain IPF.

IPF estimates (i.e., refines) the entries in the sample multiway table (in this instance, the PUMS) to make them exactly match the known marginal totals (in this case, the SF 3 summary data) while maintaining the sample table's correlation structure. The algorithm is exceedingly simple. The algorithm begins by converting all summaries and tables to proportions of the total. For example, Table 3 and Table 4 become Table 5 and Table 6, respectively. In terms of proportions, the PUMS sample for these two demographics is shown in Table 7.

Table 5. Proportion of workers in family households for census tract 1, block 2 of Los Alamos County, NM.

Proportion of Family Households, *n*, with Number of Workers in the Household				
Workers	0	1	2	>2
Prop.	0.000	0.336	0.594	0.069

[5] Deming, W.E. and Stephan, F.F. (1940), On A Least Squares Adjustment Of A Sampled Frequency Table When The Expected Marginal Tables Are Known, *Annals of Mathematical Statistics*, Vol. 11, pp 427-444.

Table 6. Proportion of ages of householders for census tract 1, block group 2, of Los Alamos County, NM.

Proportion of Family Households, *n*, with Householder Age in the Given Ranges							
Age	15-24	25-34	35-44	45-54	55-64	65-74	>74
Prop.	0.011	0.372	0.261	0.128	0.128	0.100	0.000

Table 7. Cross-classification of the proportion of workers and the age of the householder for PUMA 00400, which contains census tract 1, block group 2 of Los Alamos County, NM.

Householder Age								
Workers	15-24	25-34	35-44	45-54	55-64	65-74	>74	Total
0	0.001	0.007	0.006	0.002	0.017	0.042	0.028	0.104
1	0.077	0.072	0.081	0.032	0.053	0.040	0.012	0.297
2	0.019	0.090	0.182	0.103	0.056	0.015	0.004	0.468
>2	0.000	0.002	0.043	0.050	0.027	0.007	0.002	0.131
Total	0.027	0.170	0.312	0.188	0.153	0.104	0.046	

IPF converts the PUMS proportions in Table 7 so that they have the same row and column proportions as the SF 3 data given in Table 5 and Table 6. IPF accomplishes this by first changing the rows then the columns according to the following rules:

- Update the first row of Table 7 by multiplying each entry by the first marginal proportion for that row given in Table 5 and dividing by the total for that row on the last iteration. In this case, the first element of the first row of Table 7 becomes 0.001*0.000/0.104=0.000.

- This process continues with the remainder of the rows of Table 7, where (for example) the third entry of the second row becomes 0.081*0.336/0.297=0.092.

After all the rows are updated, the same procedure is applied to each column. The procedure continues by alternating between rows and columns until the table entries no longer change. For tables with more than two dimensions, the same procedure is followed—updating one dimension at a time. Table 8 shows the final results of this procedure, based on the data in Table 7.

If required, the forecast procedure updates these tables. In either case, the last step in household generation is to draw samples from the PUMS. There are 360 family households in the block group given below. For this block group, 360 households are generated—one at a time—following this procedure:

- First, a category of age and the number of workers are selected at random according to the probabilities in Table 8.

- And second, given the category (e.g., a householder between 45 and 54 years of age in a household with two workers), one of the households in the PUMS matching these demographics is drawn at random. In this case, one household would be drawn from the 156 households possible (as shown in Table 4).

This process is repeated 360 times to form a population that matches the census. Note that the same household from the PUMS may be selected multiple times by this procedure.

Table 8. Estimated cross-classification of the proportion of workers and the age of the householder for families in census tract 1, block group 2 of Los Alamos County, NM.

	Householder Age							
Workers	15-24	25-34	35-44	45-54	55-64	65-74	>74	Total
0	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
1	0.003	0.141	0.061	0.020	0.047	0.063	0.000	0.336
2	0.009	0.228	0.178	0.086	0.065	0.030	0.000	0.594
>2	0.000	0.003	0.022	0.022	0.016	0.007	0.000	0.069
Total	0.011	0.372	0.261	0.128	0.128	0.100	0.000	

Tables 1, 3, and 4, as well as the paper by Beckman, Baggerly, and McKay[6], show that fitting only one block group at a time using IPF is not entirely correct. IPF is based on the assumption that the seed proportions (as given by the PUMS, Table 7) are a sample of the population that produces the exact marginal totals given by SF 3 (shown here in Table 7). Table 1 shows there are no households with 0 workers in the block group, while Table 4 shows many 0-worker households.

The PUMS consists of a sample of households that contain all or parts of multiple block groups. In this case, block group 2 of census tract 1 from Los Alamos County is just one of the many block groups in PUMA 00400. PUMA 00400 contains all of the block groups in Los Alamos and Santa Fe counties of New Mexico. That PUMA 00400 is a sample of multiple block groups is evident from PUMS.

4.4 Creating the Population

The Population module uses the following steps to construct a true synthetic population. A two-step procedure is used to modify the IPF routine so that it can simultaneously consider all block groups that make up a PUMA. A final step is added to take into account the forecast marginal inputs.

Step One Assemble each block group in a PUMA from the MABLE/Geocorr2k database.

Step Two Collect the marginal SF 3 tables for the block groups in the PUMA.

Step Three Construct from the PUMS a multiway demographic table that matches the demographics from the SF 3 tables for the corresponding PUMA. The entries of this table are the sums of the household weights from the PUMS.

[6] See Footnote 3.

Step Four

- Add the marginal tables for all of the block groups in the PUMA.
- Estimate a multiway table for the entire PUMA by using an IPF fit of this summed table to the PUMS.

Step Five

- Use the estimated table as an additional marginal table.
- Create an (*m+1*)-dimensional table.

The first *m* dimensions are the *m* marginal totals from SF 3, whereas the $m+1^{st}$ marginal is created by *stacking* all of the marginal tables. This (*m+1*)-dimensional stacked table, along with the table estimated from the sums, are the marginal tables used in an IPF procedure to an (*m+1*)-dimensional table consisting entirely of ones. This results in an estimated multiway table for each block group in the PUMA. More information on this process can be found in the paper by Beckman, Baggerly, and McKay[7].

In a forecast setting, Step Six is executed; otherwise, it is skipped.

Step Six

- Combine the block group estimates from the step above to form one multidimensional array.
- Using iterative proportional fitting, fit sets of forecast marginal totals against the multidimensional array—one block group at a time.

Forecasted populations are generated using a modification of the procedures given in the paper by Beckman, Baggerly, and McKay[8]. The technology's user must supply the algorithm with the SF 3 and PUMS data for the base year as required by Beckman, Baggerly, and McKay, which is shown in the five steps above. Additionally, the user must supply the same set of summary demographics used above for the households in each block group in the projected year.

To forecast a population given projected SF 3-type marginal data on a census block group basis for the entire region, one starts with an individual PUMA. At the end of Step Five, a multidimensional array has been estimated for each census block group in the PUMA. The dimensions of the array represent particular demographic types. The entries in the array are the proportions of households of the various demographic types generated in the complete baseline procedure.

[7] See Footnote 3.
[8] See Footnote 3.

The forecast procedure proceeds as follows. The block groups in the PUMA that have the fewest zero marginal totals are identified. Of these, the block groups with the maximum number of non-zeros in the estimated proportion arrays are identified. Of these, the block group with the largest number of households is selected. The proportions from this block group are the best available representation of the PUMS sample and will take the place of the PUMS sample during forecasting.

To compute a set of cross-classified proportions for each block group in the PUMA, we use the representative proportions identified above together with the forecast marginal totals provided by the user and perform iterative proportional fitting as described by Beckman, Baggerly, and McKay[9]. Households will be selected from the PUMS according to these proportions as described in the Beckman, Baggerly, and McKay[10] paper.

Step Seven Draw random households from the PUMS that match the demographics of each of the cells of the estimated multiway table for each block group. The number of households generated equals the total number of households in the marginal totals for that block group.

4.5 Adjustments to the IPF Routine

Minor adjustments must be made to the IPF routine to handle marginal summaries in table form. For example, two-way marginal totals are considered to be one marginal by the IPF routine. Such marginal tables are converted to a single demographic whose categories consist of all the combinations of the two demographics involved.

If two marginal tables contain a common demographic variable, the procedure is not altered and the fitting proceeds as above, treating each marginal separately.

In cases in which one demographic variable is in two summary tables (a one-dimensional table and a two-dimensional table) and has fewer categories in the two-dimensional table, an additional step is required. The procedure uses only one marginal table at a time. When the table with the *collapsed* marginal is considered, the procedure updates the cells as usual where all of the cells that contribute to the individual collapsed categories are updated by the same proportion.

4.6 Locating and Numbering the Population; Assigning Vehicle Emissions Type

The baseline synthetic population is produced on a block-group basis—no other information about the location of individual households is known. To place each

---

[9] See Footnote 3.
[10] See Footnote 3.

household on the transportation network, procedures are developed using land-use data. The number of vehicles owned by each household is given in the PUMS and is, therefore, in the synthetic population.

The Emissions Estimator module requires that the vehicles be identified by emissions type. However, the emissions type of the vehicles in the household is unknown. The Population module contains a model to assign vehicle types to each vehicle in the population. The last step in the creation of the synthetic population is to assign a unique number to each household and each person in the population.

4.6.1 Activity Location

Each household in the population is located at an activity location on the transportation network. These are usually on the *walk* portion of the network.

Each network is required to have an *Activity Location* file. This file contains the locations of those places on the network in which activities may take place. Associated with these locations is a set of land-use characteristics that indicate the type of activities that may take place at that location.

Each network has a unique set of land-use characteristics associated with its activity locations. Land use is used to form a weighting factor for each activity location that represents the relative likelihood of a housing unit being placed there. The exact formulation of these weights depends on the network under investigation and the availability of land-use information. For a network representing a real metropolitan area, the land use could, for example, contain the square footage of single-family residential housing along with the square footage of multi-family housing that surrounds the activity location.

In this case, the weights for the activity locations could be formed by adding the square footage of single-family residential housing to a multiple of the square footage of multi-family housing.

4.6.1.1 Placing Households on Activity Locations

Given the housing weights associated with each activity location on the network, households are placed on these locations by taking the steps outlined below.

Step One

- Identify all activity locations within a block group.
- Assume that there are n of them.
- Denote the associated household weights for these activities by w_I .
- Compute the probabilities as follows: $p_i = w_i/\Sigma w_I$.

Step Two

- Assign each individual household in the block group to one of the n activity locations according to the probabilities, p_I.
- The location of the households is one of the required demographics for each synthetic household.

4.6.2 Vehicles and Their Emissions

Each synthetic household is created with a number of vehicles assigned to it. These vehicles have a unique number and are identified as belonging to the household.

Assigned to each vehicle in the population is one of the vehicle emission types, which are described in Volume Three (*Modules*), Chapter Seven (*Emissions Estimator*). This assignment may be done at random according to either a national or local distribution of vehicle emission types.

Each vehicle is also assigned a starting location, which consists of one of the parking locations on the driving network. Traditionally, this location has been the parking location closest to the household location. This information is written to the vehicle file.

4.6.3 Assigning Identification Numbers

The final step in generating a synthetic population is to assign a unique number to each household and person in the population. The person number is a unique identifier carried through the Route Planner to the Traffic Microsimulator. All output that is person-oriented references these numbers.

Each vehicle driver must have an entry in the synthetic population. Therefore, fictitious individuals are added to the population to represent those who travel on the network but do not live in the area undergoing study. Known as "itinerant travelers," these are added to the synthetic population as single-person households, each with one vehicle. The same is true for transit drivers and freight haulers.

These households, along with the individual, are given their own unique household and person number. If demographics are added to the actual synthetic persons or households, the same demographics must be added to the itinerant traveler population. In some cases, these may be meaningless numbers because the activity list for these travelers is generated from origin-destination tables independent of the individual's demographics.

In equity studies, itinerant travelers can be viewed as a separate population. Every itinerant traveler owns one vehicle. The vehicle is given a unique number and type and is placed in the vehicle file. The starting location of these vehicles is the parking location where the traveler's trip begins. These starting points are most likely on the boundary of the study area, as itinerant travelers are those that are passing through the area or entering the area from the outside.

5. Using the Population Generator

5.1 Overview

The Population module contains several programs that are executed in order. Population and vehicle files are prepared for use by other UIS applications.

The Population Generator expects input files that are defined with the specific formats described in Section 7. The Microsoft Access database management system may be used to facilitate preparation of files that contain data obtained from the Census Bureau. A skeleton database plus several Perl scripts are provided to aid the user in preparing valid input files. Use of these tools is not required; any methods that produce files in the required format may be used.

5.2 Assembling the Required Input Data

Before setting out to generate a synthetic population, the user must first decide which household demographics will be used in the process. Use the Summary File 3 documentation[11] to identify which summary file segments contain the chosen demographics. In the following, it is assumed that the selected demographics are household size, householder age, household income, and vehicles available. Summaries for these demographics are contained in summary file segments 1, 6, and 58.

Data from the Census Bureau is available for downloading via the Internet or for purchase on CD-ROM. The following procedures describe how to download the data.

Step One Create a data directory for each state in the metropolitan area where Census Bureau data will be stored by state.

Step Two Download Summary File 3 (SF 3) files from the following Census Bureau site:
http://www2.census.gov/census_2000/datasets/Summary_File_3

Select a state from the list.

(In the following, st represents the 2-character abbreviation for the state.)
Select st00001_uf3.zip and save it to disk in the data directory.
Select st00006_uf3.zip and save it to disk in the data directory.
Select st00058_uf3.zip and save it to disk in the data directory.
Select stgeo_uf3.zip and save it to disk in the data directory.

[11] See Footnote 1.

Step Three In the data directory, unzip each of the files.

The *uf3* suffix is not recognized by Access. Rename each file, changing *.uf3* to *.txt*.

Step Four Download Public Use Microdata Sample (PUMS) files from the following Census Bureau site:
http://www2.census.gov/census_2000/datasets/PUMS/FivePercent

Select a state from the list.

Select *all_st.zip* and save it to disk in the data directory.

Step Five In the data directory, unzip the file to produce *PUMEQ5-st.TXT* and *PUMS5_n.TXT*, where *n* is the number of the state.

Step Six Use the Perl script *SplitPums.pl* to divide the PUMS data into two files, one containing all of the households and the other containing all of the persons.

```
perl SplitPums.pl PUMS5_n.TXT PumsH.txt PumsP.txt
```

Step Seven Construct and download a *.csv* file from the MABLE/Geocorr2K site:
http://mcdc2.missouri.edu/websas/geocorr2k.html

Select state to process

For SOURCE Geocode, select: PUMA for 5 Pct Samples (2000)

For TARGET Geocodes, select: State (2000) AND Census Block Group (2000)
(Use Ctrl key to make second selection.)

For Weighting Variable, select: Population (2000 census)

Do NOT check the box Ignore Census Blocks.

For Output Options, select: Generate a CSV file AND Codes and Names

Run Request

Select the comma-delimited ("csv") file and save it to disk in the data directory.

Step Eight Verify that the downloaded file contains the following columns in this order:
"puma5", "state", "county", "tract", "bg", "stab", "cntyname", "pop2k", "afact"

Step Nine Use the Perl script *EditGeocorr.pl* to modify fields in the Geocorr data that differ slightly from corresponding fields in the census data.

```
perl EditGeocorr.pl geocorr2k.csv Geocorr.txt
```

The resultant Geocorr.txt file will have the columns expected by the Access import specification in this order:
PUMA, STATE, COUNTY, TRACT, BLKGRP, POP2K

5.3 Using Access to Manipulate Census Bureau Data

5.3.1 Importing Data into Tables

Step One Copy */data/Population/skeleton.mdb* to a new database that will be used to contain the data for one state in the metropolitan area. Open this database using Access.

Step Two Import data tables into the database using the following procedure:

1) On the File menu, point to Get External Data, and click Import
2) In the Files Of Type list, select Text Files from the menu
3) Locate the data directory and select *st00001.txt*
4) Click Import
5) In the Import Text Wizard, click Advanced...
6) In the Import Specification, click Specs...
7) Select the SF300001 Import Specification from the menu and click Open
8) In Import Specification window, click OK
9) In Import Text Wizard, click Next>
10) Click Next> again
11) Store data In an Existing Table and select SF300001 from the menu
12) Click Next>
13) Click Finish

Step Three Define a primary key for the table.

Open the table in Design view.

Right click on LOGRECNO and select Primary Key.

Close the table, saving changes to its design.

Repeat Steps Two and Three for *st00006.txt* using the Import Specification SF300006 and storing the data in existing table SF300006. Set LOGRECNO as the primary key.

Repeat Steps Two and Three for *st00058.txt* using the Import Specification SF300058 and storing the data in existing table SF300058. Set LOGRECNO as the primary key.

Repeat Step Two for *stgeo.txt* using the Import Specification SF3GEO and storing the data in existing table SF3GEO. Do not set a primary key.

The other existing SF3 tables may remain empty.

Repeat Step Two for the PUMS households file (*PumsH.txt*) using the Import Specification PumsH and storing the data in existing table PumsH. Do not set a primary key.

Repeat Step Two for the PUMS persons file (*PumsP.txt*) using the Import Specification PumsP and storing the data in existing table PumsP. Do not set a primary key.

Repeat Step Two for the modified Geocorr file (*Geocorr.txt*) using the Import Specification Geocorr and storing the data in existing table Geocorr. Do not set a primary key.

5.3.2 Correcting Geocorr Data

Experience has shown that the data obtained from MABLE/Geocorr2K is not always self-consistent and does not always agree with data obtained from the Census Bureau. We take the Census Bureau data as the reference and adjust the Geocorr data to match it.

The Census Bureau guideline for defining PUMAs requires that a tract must be wholly contained within a PUMA. The **Check Geocorr** query is used to determine whether the same county and tract appear in more than one PUMA. When this occurs, the Geocorr table is modified to assign all block groups in the tract to the PUMA in which the tract has the larger recorded population. Since the population counts from Geocorr are not used anywhere, the record with the smaller population may simply be deleted from the Geocorr table.

Step One Modify the **Check Geocorr** self-join query to specify criteria for each of the PUMA columns in the table. The first column is specified as equal to a particular PUMA, and the second PUMA column is specified as not equal (<>) to that PUMA.

The query stored in *skeleton.mdb* contains the value “00100” as an example. In Design view, modify this value to a PUMA of interest, taking care to include any leading zeros in this text value.

Step Two Run the query.

If a non-empty table is returned, one or more block groups appear in two PUMAs.

When this occurs, execute Step Three.

Step Three Open the Geocorr table.

The Find button may be successively used to locate all occurrences of the tract number in the Geocorr table.

Type the tract number in the Find What field.

From the Look In list, select Geocorr : Table.

From the Search list, select All.

Identify the record to be deleted according to the population criterion, and use the Delete Record button to delete the selected record. Confirm the deletion.

When the PUMA to which the tract is assigned does not contain entries for all of the block groups in the tract, records with zero population should be added for missing block groups so that all block groups of each tract are assigned to a single PUMA.

Repeat Steps One and Two for each PUMA of interest, deleting any duplicate records that are revealed from the Geocorr table as described in Step Three.

The consistency of the Geocorr and SFGEO tables may be checked by running the **County BlockGroups Geocorr** and **County BlockGroups SFGEO** queries.

Step One Modify the County criteria in each of these queries to specify the number identifier of the county you want to compare.

Step Two Run the queries.

Both queries should return the same number of records. If not, the tables are inconsistent and must be corrected to be consistent.

5.3.3 Other Tables

In order to compute the indexes for the demographics used in IPF, the bounds for each of the levels must be known. Storing these bounds within the Access database permits the indexes to be computed and written by the same queries that extract the household characteristics. The number of levels is generally small, so composing the tables is straightforward. The number of levels to use is chosen by the analyst based on the breakdown of data available in the summary tables. Several levels available in the summary data may be aggregated into a single level for the purpose of population generation.

The skeleton database contains tables for Age Levels, Income Levels, Size Levels, and Vehicle Levels. The PUMS queries described in the next section use the Levels information to assign indexes to each household.

5.3.4 Description of Queries

Queries are used to extract relevant subsets of information from larger data sets. Criteria control how information is extracted and presented. The queries in *skeleton.mdb* that extract demographics from the SF 3 data by block group are **Household Income, Household Size, Householder Age,** and **Vehicles Available.** As written, each query extracts data for the whole state. These queries may be used as examples if other household demographics are to be used to generate a synthetic population.

The Puma field or the County field may be given criteria if the user prefers to extract data for only part of the state. The disadvantage of partial extraction is that more files must be dealt with. However, if the population is to be generated in parallel, it may be advantageous if the parallel processes are accessing separate files rather than simultaneously attempting to access a single file.

The queries that extract PUMS data and provide the indexes required by the Population Generator are also included in *skeleton.mdb*. The **PUMS Households** query includes the SERIALNO, PUMA5, HWEIGHT, PERSONS, VEHICL, HINC, P18, HHT, and WORKERS demographics, as well as the indexes I-INCOME, I-SIZE, I-AGE, and I-VEHICLE. The **PUMS Persons** query includes the RELATE, SEX, AGE, ESR, OCCCEN5, TRVMNS, and TRVTIME demographics. Additional demographics may be added to either query as desired; refer to the PUMS documentation for descriptions of the available demographics.[12] The Population Generator handles any number of demographics in an arbitrary order.

The **PUMS Households** query relies on the **Pums Temporary** sub-query which in turn relies on the **Pums Householder Age** sub-query to extract householder age and the **Pums Household Workers** sub-query to extract the number of workers in a household from the PumsP person table. The **PUMS Persons** query uses the **PUMS Households** query to associate persons with their respective households. The PUMS queries take longer to run

[12] See Footnote 2.

than the SF 3 queries because of the need to nest queries as described. As written, each query extracts data for the whole state.

The Puma5 field may be given criteria if the user prefers to extract data for specific regions. Alternatively, the Puma1 field (super-puma) may be added, preferably hidden, and given criteria.

5.3.5 Exporting Query Results

The information retrieved by running a query may be exported into flat files and used as input by programs such as the Population Generator. The following steps describe the procedure for exporting query results.

Step One Select the Household Income query by clicking on it.

Step Two Export the query using the following procedure.

On the File menu, click Export…

In the Save As Type list, select Text Files from the menu

Locate the data directory where the file will be saved.

Choose a file name for the exported file; default is the query name.

Click Export

In Export Text Wizard, select Delimited format

Click Next>

For the delimiter, select Space

Select the Include Field Names on First Row check box

For the Text Qualifier, select {none} from the menu

Click Next>

Click Finish

Repeat these steps for the Household Size query.

Repeat these steps for the Householder Age query.

Repeat these steps for the Vehicles Available query.

Repeat these steps for the PUMS Households query.

Repeat these steps for the PUMS Persons query.

5.4 Prepare Input Files for Population Generation

5.4.1 Adjusting summary data for different universes

When SF 3 queries use tables with different population universes, an adjustment must be made so that all queries pertaining to a block group have the same marginal totals. For example, the **Vehicles Available** query uses table H44, which has a universe of occupied housing units, while the other three SF 3 queries use tables that have a universe of households. In order to be combined with the other demographics for population generation, the vehicles available must be adjusted to conform to the other tables. The program *ConformSummary* may be used for this task.

```
ConformSummary <config-file>
```

The required configuration file keys are described in Appendix B.

5.4.2 Combining query results into input files

The files containing SF 3 query results must be combined into a single file in the format required for summary files. The Perl script *CombineSummary.pl* may be used for this task.

```
perl CombineSummary.pl <file1> ... <fileN> Summary2000
```

The first filenames are the exported query files. The script handles an arbitrary number of queries, and the order in which the files are combined must match the order specified by `POP_TABLES` in the configuration file. Refer to Section 5.5.1 for more information about the configuration file.

The final filename is the name of the output summary file. All files must contain the same count of total entries for each block group.

The PUMS file subsets selected by the queries for households and persons must be recombined into a single sample file containing each household record followed by the person records for that household. The Perl script *CombinePums.pl* may be used for this task.

```
perl CombinePums.pl Pums_Households.txt Pums_Persons.txt Sample2000
```

The first two filenames are the exported query files. The last filename is the name of the output sample file.

5.5 Running the Population Generator

5.5.1 Preparing the configuration file

The Population Generator uses configuration file keys to control generation of the population. Configuration sets are used to define related configuration file keys that are used to specify inputs to the IPF procedure. The set `POP_DEMOGRAPHICS` defines the demographics used in IPF, and the set `POP_TABLES` specifies how these demographics appear in summary tables.

Each subset of the `POP_DEMOGRAPHICS` set defines a demographic in `POP_DEMOGRAPHIC` and the associated number of levels for the demographic in `POP_LEVELS`. Define a subset for each demographic variable to be used in IPF. Typically, the subset names simply number the subsets from 1 to N demographics.

Each demographic to be used in IPF must appear in one or more of the marginal summary tables. The layout of the tables is specified using the `POP_TABLES` set. Each table that will be used in IPF is defined as a subset of `POP_TABLES`. Typically the subset names simply number the subsets from 1 to M tables. Be careful to list the subsets in the same order as they appear in the Summary file.

The demographics included in the table are named in `POP_TABLE`. If the table is multidimensional, the demographic names are separated by a semicolon. The splits to be used are defined with `POP_SPLITS`. If each demographic level corresponds to a split, the splits can be inferred and the `POP_SPLITS` key is optional. However, if levels are to be combined, the `POP_SPLITS` key is required to define which levels are combined. The splits are whitespace delimited. If the table is multidimensional, the splits for each dimension are whitespace delimited, and the dimensions are separated by a semicolon. The user-specified levels and splits are checked for consistency by the Population Generator.

Appendix J provides some examples of configuration sets and their meanings within the Population Generator.

Other configuration file keys that are required include `POP_SAMPLE_FILE`, `POP_BASELINE_SUMMARY_FILE`, `POP_BASELINE_POPULATION_FILE`, and `POP_PUMA`. Appendix A provides an explanation of all the configuration file keys used by *GeneratePopulation*, including those keys for which default values exist.

5.5.2 Generating the population

To generate a population, use the following command line:

```
GeneratePopulation <config-file>
```

The baseline population will be recorded in the file specified by the configuration file key `POP_BASELINE_POPULATION_FILE`. If a forecast population is requested by providing a value for the `POP_FORECAST_POPULATION_FILE` configuration file key and a forecast summary file is furnished, the forecast population will also be generated and recorded in `POP_FORECAST_POPULATION_FILE`. The log file that is produced should always be examined to confirm that no problems were encountered.

5.6 Locating a Population on a Transportation Network

The *BlockGroupLoc* utility generates home locations for populations on a transportation network by correlating census tract and block group user data values specified in the network activity location file with tract and block group data in the baseline population. Candidate home locations must have the same state, county, census tract, and block group as the household; they also must have residential land-use values greater than zero.

Some households may be in block groups that do not have any activity locations on the transportation network that are associated with that block group. Alternative tract and block groups may be specified for these households. The alternative state/county/tract/block group tuples are specified in a Tract/Block Group Substitution file, defined below.

BlockGroupLoc also generates household and person IDs and assigns them to the located population. The user data in the activity location table in the transportation network must contain state, county, tract, block group, and residential and commercial land use values.

To locate a population, use the following command line:

```
BlockGroupLoc <config-file>
```

BlockGroupLoc uses configuration file keys specified in a configuration file. Some keys have default values that may be used if the key is not specified in the configuration file. Appendix C lists these configuration file keys.

5.7 Generating a Vehicle File for Located Synthetic Populations

A vehicle file contains information about the initial locations of a household's vehicles. For most households, the starting location of the vehicle will be the parking location near the household's home location.

Vehgen is a utility that creates a vehicle file containing information about the household's vehicles and their starting locations.

Each vehicle's starting parking location is found by iterating through the process links connected to the home activity location. Every home activity location must have at least one parking location accessible via the activity location's process links.

To generate a vehicle file, use the following command:

```
Vehgen <config-file>
```

Vehgen uses configuration file keys specified in a configuration file. Some keys have default values that may be used if the key is not specified in the configuration file. Appendix D lists these configuration file keys.

6. POPULATION VISUALIZATION

Various characteristics of the located population may be visualized with the UIS Output Visualizer. For example, one may visualize household characteristics such as number of persons or number of vehicles at each home location. Several auxiliary programs are provided that take population and/or activity files and prepare files formatted as network feature files for use as input to the Visualizer. In the following discussion, *PrepareVisualFile2, PrepareVisualFile4,* and *SamplePopulation* are generic programs that work with any population, whereas *PrepareVisualFile1* and *PrepareVisualFile3* require specific characteristics to be present in the input files.

6.1 PrepareVisualFile1

This program prepares a network feature file that records a count of the number of households at each home location. It also records the average number of vehicles, number of persons, number of workers, and average household income at each home location. The features that may be recorded depend on the population characteristics that were retained when the population was generated.

To prepare this visualization file, use the command:

```
PrepareVisualFile1 <config-file>
```

Appendix E lists the configuration file keys used by this program.

6.2 PrepareVisualFile2

This program prepares a network feature file that records a count of the number of households at each home location that have no vehicle.

To prepare this visualization file, use the command:

```
PrepareVisualFile2 <config-file>
```

Appendix F lists the configuration file keys used by this program.

6.3 PrepareVisualFile3

For each activity location, this program prepares a network feature file that records the number of persons having an activity of some type at that location. The types of activities that are tallied depend on the activity types identified in the activity file. A person performing the same activity at the same location multiple times is counted only once.

To prepare this visualization file, use the command:

```
PrepareVisualFile3 <config-file>
```

Appendix G lists the configuration file keys used by this program.

6.4 PrepareVisualFile4

This program prepares a network feature file that sequentially exhibits the difference between two activity sets. A count of the number of activities occurring at an activity location is recorded in uniformly-sized time bins along with the difference in the count compared to a baseline activity set. The resulting file is then sorted in ascending time order for sequential display by the Visualizer.

To prepare this visualization file, use the command:

```
PrepareVisualFile4 <config-file>
```

Appendix H lists the configuration file keys used by this program.

6.5 SamplePopulation

This program randomly constructs a sample of a located population and writes the sample population and the activities associated with the households in the sample to new files. The sample is drawn from a single county that is specified by the user. The fraction of the population to include in the sample is also controlled by the user.

To extract a sample of the population, use the command:

```
SamplePopulation <config-file>
```

Appendix I lists the configuration file keys used by this program.

7. FILES

This section describes the formats for the required input files and the output files produced by the Population Generator.

7.1 Sample File

The sample file contains household and person demographic information that may be extracted from a PUMS or obtained elsewhere. The entire PUMS record need not be supplied, only the required fields for the IPF and population output. The fields in this file are whitespace-separated integers.

For household demographics used in IPF, the household record also contains an index for each demographic that gives the level within the demographic for the household.

The file contains two header lines—one for households and one for persons. The header line lists the names of the demographics in the data records. For the demographic indexes used in the IPF, the name must begin with "I-".

<u>Example</u>:

```
#RECTYPE SERIALNO PUMA5 HWEIGHT PERSONS VEHICL HINC P18 HHT WIF I-INCOME I-SIZE I-AGE I-VEHICLE
#RECTYPE SERIALNO PNUM RELATE SEX AGE ESR OCCCEN5 TRVMNS TRVTIME
H 142 03409 21 3 1 1200 0 3 1 1 3 5 2
P 142 1 01 2 64 6 000 00 000
P 142 2 03 2 26 3 992 00 000
P 142 3 03 1 30 6 000 00 000
```

This example shows the header lines followed by a household containing three persons. The household and its members share a SERIALNO of 142. The household appears in PUMA 03409 and owns one vehicle. The three-person records follow the household record and contain numerous person demographics.

7.2 Summary File

The summary file contains the marginal tables used by IPF. For readability, the file contains an optional header that describes the contents of the file. The first word of the header must be "`#Summary`:". The rest of the header is arbitrary and contains descriptive information supplied by the user.

<u>Example</u>:

```
#Summary: Householder Age, Household Income, Household Size, Household Vehicles
17 031 03501 010100 1 2238
 173 675 668 361 213 97 51
 468 175 372 348 317 412 146
 919 584 284 215 133 78 25
 1020 889 306 10 13 0
```

This example shows a header line followed by the records for a single block group.

- The first record contains the state (17), the county (031), the PUMA (03501), the tract (010100), the block group (1), and a count of the number of households in this block group (2238).
- The second record contains the household counts for each of seven levels of householder age.
- The third record contains the household counts for each of seven levels of household income.
- The fourth record contains the household counts for each of seven levels of household size.
- The fifth record contains the household counts for each of six levels of vehicles per household.

Note that the counts for each demographic record must sum to the value of the count in the first record (2238 in this example).

The tables for each block group must appear in the order specified by the `POP_TABLES` subsets in the configuration file. Each table begins on a new line. A table may use several lines. The number of values in the table will be compared with the splits to ensure that the correct number of values is read.

7.3 Temporary files used by IPF

The temporary files used by IPF are created in the directory specified by `POP_IPF_FILE_DIRECTORY`.

The sample input to IPF will be assembled from the household indexes data and the configuration information.

The summary input to IPF will be assembled from the summary tables by block group and the configuration information.

The output from IPF will be read and used to generate a population.

The temporary files used by IPF may be retained by specifying the true value for the configuration key `POP_KEEP_TEMP_FILES`.

7.4 Population File

The synthetic population file produced by the Population Generator contains three header lines followed by the data records that define the households and persons in the population.

7.4.1 Header Lines

All of the header lines begin with the '#' character. The first line of the file indicates the version number of the population file. The second line contains several fields that identify the household and its location as well as the names of the household demographics in the population. The third line identifies each person and contains the names of the person demographics in the population.

In both the household and person demographic headers, several required fields appear first, followed by a colon (:). A single word description of each of the optional demographics in the file follows the colon.

7.4.2 Data Lines

The data for a single synthetic household spans multiple lines in the population file. The first line contains information about the household. The required fields in the household record include the state, county, tract, and block group where the household is located, followed by the character 'H', followed by a unique household identifier, the number of persons in the household, the number of vehicles in the household, and the activity location identifier of the household's home location. The optional household demographics follow the required fields.

When the population is generated, the household identifier and home location have the value -1. The *BlockGroupLoc* program, described above, may be used to assign a unique household identifier and a home location to the household. Every household must be assigned a home location before using UIS modules.

Following the household data are N lines of person data, where N = the number of persons in the household. The required fields in the person record include the household identifier, the character 'P', and the person identifier. These fields are followed by the optional person demographics. Both the household identifier and the person identifier initially have the value -1 as a placeholder, and may be assigned using *BlockGroupLoc*.

7.4.3 Example

```
#VERSION 2
#State County Tract BlkGrp H HId PERSONS VEHICL HLoc : SERIALNO HINC P18 WIF
#HId P PId : AGE RELATE SEX ESR OCCCEN5 TRVMNS TRVTIME
17 31 803603 1 H -1 4 1 -1 2734896 91700 2 2
-1 P -1 37 1 1 1 485 11 0
-1 P -1 38 2 2 1 172 1 30
-1 P -1 10 3 1 0 0 0 0
-1 P -1 8 3 1 0 0 0 0
```

This example contains a single household composed of four persons. The household is located in state 17, county 31, tract 803603, block group 1, but a home activity location has not yet been assigned (-1). The household owns one vehicle and has a household income of $91700. Both adult members are employed. The two boys are ages 10 and 8.

7.5 Log File

A log file is produced by the Population Generator and stored in the file specified by the `LOG_FILE` configuration file key. The log file contains general information about the number of demographics and tables used in the IPF, the number of households and persons in a PUMA, the number of locations and households in the summary data, and the number of households generated. It may also contain warnings or error messages if problems are encountered during the generation process.

7.6 Tract/Block Group Substitution File

Block groups are located at activity locations on a transportation network based on the state, county, census tract, and block group associated with the activity location in the network Activity Location Table. When locating a population on a sparse transportation network, some census tract/block groups may not have any associated activity locations.

The Tract/Block Group Substitution File enables users to specify a list of alternate candidate activity locations where the households in the missing tract/block group can be located. The home location is selected from the activity locations that match the alternative locations specified in the file. The configuration file key `POP_NEAREST_BG_FILE` is used to specify the name of the Tract/Block Group Substitution File. The file format contains two header lines followed by the data lines.

7.6.1 Header Lines

The first line of the file contains the version number. The second line of the file contains the columns STATE COUNTY TRACT and BG followed by ST-1 CO-1 TR-1 BG-1 ... ST-n CO-n TR-n BG-n, where n is the number of alternate block groups that will be specified. The fields in the line are whitespace delimited.

7.6.2 Data Lines

The data lines in the file contain the state, county, tract, and block group of each block group with no activity locations followed by a list of alternate locations.

7.6.3 Example

```
#VERSION 2
STATE COUNTY TRACT   BG  ST-1 CO-1 TR-1   BG-1 ST-2 CO-2 TR-2   BG-2
17      37      1100      1     17     37     1100      2      17      37    1000      5
```

Tract 1100, block group 1 within state 17, county 37 is not found in the activity location file. Activity locations from tract 1100, block group 2 in state 17, county 37, and locations from tract 1000, block group 5 in state 17, and county 37 can be used as candidate home locations for households in tract 1100, block group 1 of state 17, county 37. It is necessary to include state and county information in the file because identical tract and block group identifiers may be used in multiple counties.

7.7 Network Feature Visualization File

The format of a Network Feature Evolution file is described in the documentation for the Output Visualizer. When using this format to visualize population and/or activity files, the Feature Type field is “a” for activity location, and the Feature IDs are all set to the ID of the same activity location. Up to 10 optional data values may be included in each record.

8. EXAMPLE

This section contains an example that illustrates the use of the Population Generator and some of the auxiliary programs and scripts that are provided with the Population module. All of the files referenced in this section are included in the */example/Population* subdirectory.

8.1 Getting Started

The data in this example are for a single PUMA in McHenry County, Illinois. The example begins at the point in the process where queries have been exported from the database into text files.

Copy the following files from /*example/Population* to a new directory, where the example population will be produced:

- *Householder_Age.txt*
- *Household_Income.txt*
- *Household_Size.txt*
- *Vehicles_Available.txt*
- *PUMS_Households.txt*
- *PUMS_Persons.txt*
- *conform.cfg*
- *genpop.cfg* xx

8.2 Running *ConformSummary*

Edit *conform.cfg*, modifying the pathnames of the files to be the locations of the files copied in the previous step, and naming a new location for the output file. Do not modify the random seed if you want to be able to compare the new output to the example output.

Run *ConformSummary* using the following command:

```
ConformSummary conform.cfg
```

Two warnings are produced complaining that the file headers do not begin with the comment character; these can safely be ignored. The new output file should be identical to */example/Population/Household_Vehicles.txt.*

8.3 Running *CombineSummary.pl*

Combine the SF3 summary files into a single file in the format required by *GeneratePopulation* using the Perl script *CombineSummary.pl* as follows:

```
perl CombineSummary.pl Householder_Age.txt Household_Income.txt Household_Size.txt Household_Vehicles.txt Summary2000
```

The filenames must appear in this order to correspond with the configuration sets defined in *genpop.cfg*. The resultant file, *Summary2000*, should be identical to */example/Population/Summary2000.*

8.4 Running *CombinePums.pl*

Combine the PUMS household and person files using the Perl script *CombinePums.pl* as follows:

```
perl CombinePums.pl PUMS_Households.txt PUMS_Persons.txt Sample2000
```

The resultant file, *Sample2000*, should be identical *to /example/Population/Sample2000.*

8.5 Running *GeneratePopulation*

Create a new subdirectory below the current directory to be used for storing temporary files used by the IPF program. In the example *genpop.cfg*, this directory is called *IPFfiles*.

Edit *genpop.cfg*, modifying the pathnames of files to be the locations of the files produced in the previous steps. Change only pathnames if you want to be able to compare the new output to the example output.

Run *GeneratePopulation* as follows:

```
GeneratePopulation genpop.cfg
```

The output file containing the baseline population for PUMA 03001 should be identical *to /example/Population/Pop_03001*. The log file should be similar to */example/Population/popgen.log* except for dates and times.

8.6 Completing the Population

Two final steps are necessary to complete the population generation. Home locations and household and person identifiers are assigned using the *BlockGroupLoc* program, and a vehicle file is generated using the *Vehgen* program. These programs require files from the transportation network and are not illustrated in this example. Refer to the discussion in Chapter 5 for more information.

Appendix A: *GeneratePopulation* Configuration File Keys

Configuration File Key	Description
`POP_SAMPLE_FILE`	The pathname of the input sample (e.g., PUMS) data file.
`POP_BASELINE_SUMMARY_FILE`	The pathname of the input summary (e.g., SF 3) data file for the baseline population.
`POP_FORECAST_SUMMARY_FILE`	The pathname of the input summary data file for the forecast population.
`POP_BASELINE_POPULATION_FILE`	The pathname of the output baseline population file.
`POP_FORECAST_POPULATION_FILE`	The pathname of the output forecast population file.
`POP_PUMA`	The list of one or more PUMA identifiers for which to generate a population. Multiple identifiers are whitespace delimited.
`POP_HOUSEHOLD_DEMOGRAPHICS`	The list of household demographics to include in the output population file. (May be a subset of the demographics in the `POP_SAMPLE_FILE`.)
`POP_PERSON_DEMOGRAPHICS`	The list of person demographics to include in the output population file. (May be a subset of the demographics in the `POP_SAMPLE_FILE`.)
`POP_IPF_EXECUTABLE`	The pathname of the iterative proportional fitting executable program. Default = *./IPF/IPF*
`POP_IPF_FILE_DIRECTORY`	The pathname of the directory in which to store the temporary files used in IPF. Default = "."
`POP_KEEP_IPF_FILES`	Whether to retain temporary files used in IPF. Default = FALSE. Files will be retained in `POP_IPF_FILE_DIRECTORY`.
`POP_RANDOM_SEED`	The integer seed used to initialize the random number generator.
`POP_TOLERANCE`	The amount of difference tolerated in the IPF results before a warning is written. Large differences may indicate a poor fit. Also, the amount a CDF is allowed to deviate from 1.0 before renormalization is performed. Default = 0.001
`POP_DEMOGRAPHICS`	The set name used by the Population Generator to group the configuration file keys that describe the demographics used in IPF.
`POP_DEMOGRAPHIC`	The name of a demographic to be used in IPF.
`POP_LEVELS`	The number of levels in a demographic used in IPF.
`POP_TABLES`	The set name used by the Population Generator to group the configuration file keys that describe the summary tables used in IPF.

Configuration File Key	Description
POP_TABLE	The name of the demographic whose marginal totals appear in the summary table. When the table is multidimensional, the demographic names are separated by a semicolon.
POP_SPLITS	Optional list of splits to be used by IPF. When not specified, each demographic level is used as a split. When the table is multidimensional, the splits for each dimension are whitespace delimited and the dimensions are separated by a semicolon.
LOG_FILE	The pathname of the log file produced.

Appendix B: *ConformSummary* Configuration File Keys

Configuration File Key	Description
`POP_REFERENCE_FILE`	A summary file from the universe being used for iterative proportional fitting.
`POP_NONCONFORMING_FILE`	The summary file from another universe that will be converted to conform to the universe of files used for IPF.
`POP_CONFORMED_FILE`	The converted summary file whose counts now match the universe used for IPF.
`POP_RANDOM_SEED`	The integer seed used to initialize the random number generator.

Appendix C: *BlockGroupLoc* Configuration File Keys

Configuration File Key	Description
ACT_BLOCKGROUP_HEADER	The user data column header in the network activity location file used to specify the block group. Default = BG
ACT_COUNTY_HEADER	The user data column in the network activity location file used to specify the county. Default = COUNTY
ACT_HOME_HEADER	The user data column header in the network activity location file used to specify single family home locations. Default = HOME
ACT_MULTI_FAMILY_HEADER	The user data column header in the network activity location file used to specify multi-family home locations. If not specified, multi-family user data from the activity location file is ignored.
ACT_STATE_HEADER	The user data column in the network activity location file used to specify the state. Default = STATE
ACT_TRACT_HEADER	The user data column header in the network activity location file used to specify the census tract. Default = TRACT
NET_ACTIVITY_LOCATION_TABLE*	The network activity location table name.
NET_DIRECTORY*	The directory where the network files reside.
NET_LINK_TABLE*	The network link table name.
NET_NODE_TABLE*	The network node table name.
POP_BASELINE_FILE*	The name of the file containing the baseline population.
POP_LOCATED_FILE*	The name of the file where the located population will be written.
POP_NEAREST_BG_FILE	The name of the Tract/Block Group Substitution file that contains information about the nearest tract/block group for block groups that have no activity locations on the transportation network.
POP_RANDOM_SEED	The random number seed (integer). Default = 985456379
POP_STARTING_HH_ID	The number from which the generated households will be sequentially numbered. Default = 1
POP_STARTING_PERSON_ID	The number from which the generated persons will be sequentially numbered. Default = 101

* Configuration file keys required for *BlockGroupLoc*. All others are optional and will use default values.

Appendix D: *Vehgen* Configuration File Keys

Configuration File Key	Description
POP_LOCATED_POPULATION_FILE	The name of the file containing the located population.
POP_STARTING_VEHICLE_ID	The number from which the generated vehicles will be sequentially numbered. Default = 100
NET_ACTIVITY_LOCATION_TABLE*	The network activity location table name.
NET_DIRECTORY*	The directory where the network files reside.
NET_LINK_TABLE*	The network link table name.
NET_NODE_TABLE*	The network node table name.
NET_PARKING_TABLE	The network parking table name.
NET_PROCESS_LINK_TABLE	The network process link table name.
NET_TRANSIT_STOP_TABLE	The network transit stop table name.
VEH_DRIVER_MINIMUM_AGE	The minimum age of a driver. Used to determine the number of persons eligible for a vehicle. Default = 16
VEH_RANDOM_SEED	The seed for the random number stream. Default = 985456379
VEH_AGE_DEMOGRAPHIC	The header that denotes the age demographic in the population file. Default = AGE.
VEHICLE_FILE	The name of the vehicle file for the population.

Appendix E: *PrepareVisualFile1* Configuration File Keys

Configuration File Key	Description
POP_LOCATED_POPULATION_FILE	The name of the file containing the located population.
POP_VISUALIZER_FILE	The name of the network feature visualization file.

Appendix F: *PrepareVisualFile2* Configuration File Keys

Configuration File Key	Description
POP_LOCATED_POPULATION_FILE	The name of the file containing the located population.
POP_VISUALIZER_FILE	The name of the network feature visualization file.

Appendix G: *PrepareVisualFile3* Configuration File Keys

Configuration File Key	Description
ACTIVITY_FILE	The name of the input file containing activities for the population.
POP_VISUALIZER_FILE	The name of the network feature visualization file.

Appendix H: *PrepareVisualFile4* Configuration File Keys

Configuration File Key	Description
`BASELINE_ACTIVITY_INDEX`	The name of the index file for an activity set containing the baseline activities for the comparison.
`ALTERNATE_ACTIVITY_INDEX`	The name of the index file for an activity set that is to be compared with the baseline set. When no alternate filename is provided, the baseline activities will be written to the network feature file.
`POP_VISUALIZER_FILE`	The name of the network feature visualization file.
`BIN_SIZE`	The size of the time bin in hours. Default = 1 hour
`TIME_MAXIMUM`	The upper bound of the time frame in which to record data. Default = 28. (The lower bound is always 0.)

Appendix I: *SamplePopulation* Configuration File Keys

Configuration File Key	Description
`POP_LOCATED_POPULATION_FILE`	The name of the input file containing the located population.
`POP_SAMPLE_FILE`	The name of the output file where the population sample will be written.
`ACTIVITY_FILE`	The name of the input file containing activities for the population.
`ACT_SAMPLE_FILE`	The name of the output file containing the activities of the people in the population sample.
`POP_STATE`	The integer identifier of the state where the sample will be drawn from.
`POP_COUNTY`	The integer identifier of the county where the sample will be drawn from.
`POP_SAMPLE_FRACTION`	The decimal fraction of the population to be sampled.
`POP_RANDOM_SEED`	The integer seed used to initialize the random number generator.

Appendix J: Configuration set examples

Example 1 contains the POP_DEMOGRAPHICS and POP_TABLES configuration sets that correspond to the example presented in Section 5. All of the tables are one-dimensional and have the number of splits equal to the number of levels, so the splits can be inferred.

```
set POP_DEMOGRAPHICS 1 active
{
POP_DEMOGRAPHIC  AGE
POP_LEVELS       7
}
set POP_DEMOGRAPHICS 2 active
{
POP_DEMOGRAPHIC  INCOME
POP_LEVELS       7
}
set POP_DEMOGRAPHICS 3 active
{
POP_DEMOGRAPHIC  SIZE
POP_LEVELS       7
}
set POP_DEMOGRAPHICS 4 active
{
POP_DEMOGRAPHIC  VEHICLE
POP_LEVELS       6
}

set POP_TABLES 1 active
{
POP_TABLE        AGE
}
set POP_TABLES 2 active
{
POP_TABLE        INCOME
}
set POP_TABLES 3 active
{
POP_TABLE        SIZE
}
set POP_TABLES 4 active
{
POP_TABLE        VEHICLE
}
```

Example 2 contains `POP_DEMOGRAPHICS` and `POP_TABLES` configuration sets that illustrate the use of a two-dimensional table and splits not equal to the number of levels. In this case, the splits must be specified.

```
set POP_DEMOGRAPHICS 1 active
{
POP_DEMOGRAPHIC  INCOME
POP_LEVELS       7
}
set POP_DEMOGRAPHICS 2 active
{
POP_DEMOGRAPHIC  AGE
POP_LEVELS       7
}
set POP_DEMOGRAPHICS 3 active
{
POP_DEMOGRAPHIC  RACE
POP_LEVELS       5
}

set POP_TABLES 1 active
{
POP_TABLE        AGE
}
set POP_TABLES 2 active
{
POP_TABLE        INCOME
POP_SPLITS       3 1 1 2
}
set POP_TABLES 3 active
{
POP_TABLE        RACE; INCOME
POP_SPLITS       1 1 1 1 1 ; 3 1 1 1 1
}
```

Appendix K: *BlockGroupLoc* Error Codes

Error codes for the *BlockGroupLoc* program are in the range 24000 – 24999.

Code	Description
24001	Caught signal.
24002	Assertion failed.
24003	Exception has occurred.
24004	Network exception has occurred.
24005	Unknown exception has occurred.
24006	Invalid program usage.
24007	Failed to open file for reading.
24008	Failed to open file for writing.
24009	Mandatory configuration key not specified.
24010	Failed to read record from file.
24011	Memory allocation failed.
24012	Mandatory file not specified.
24013	Failed to construct network.
24014	No user data for specified header in network activity location table.
24015	Insufficient data in network table.

Appendix L: *Vehgen* Error Codes

Error codes for the *Vehgen* program are in the range 33000 – 33999.

Code	Description
33001	Caught signal.
33002	Assertion failed.
33003	Exception has occurred.
33004	Network exception has occurred.
33005	Unknown exception has occurred.
33006	Invalid program usage.
33007	Failed to open file for reading.
33008	Failed to open file for writing.
33009	Failed to write record to file.
33010	Mandatory file not specified.
33011	Failed to construct network.
33012	Person demographic header not specified.
33013	Specified person demographic not found in population.
33014	Home activity location not found in network activity location table.
33015	Failed to read file header line(s).
33016	No parking location accessible via process links from specified activity location.